Time of Flight Admem Modifications

T. Shaw
Oct 1, 2001

Introduction

The Time of Flight (TOF) system utilizes ADMEM modules which have a different firmare programming of the Xilinx FPGA Pipeline chips. The TOF ADMEM Pipeline FPGAs do not form a trigger sum, rather they output threshhold comparison bits.

Implementation

Each Pipline FPGA (see Figure 1) receives four channels worth of data. Channels A and C, represent data from CAFÉ modules, Channels B and D receive data from TOFE modules.

Data from Channels A and C are fed into a threshold comparison block. This block performs the following functions:

- Read in the 16 bits of data coming from the Café module.
- Subtract an 8 bit pedestal (programmed by user and the same for both channels).
- If the MSB is set, multiply the number by 8 - all numbers are now 18 bits.
- Drop the 2 least significant bits of the number – we now have 16 bits.
- Compare the data to a low and high threshold, set a bit if the number is greater than the threshold.
- Send these comparison bits out the Trigger Sum bit path. (These bits are then sent into the trigger sum user tables for the user to massage.)

The below Table illustrates the mapping between the derived 4-bit "Raw Threshold Bits" and the look-up table address bits.

Flash RAM Look-up Table Address Bit	Signal Definition
0	0 – if Channel A*<= Threshold 1 1 – if Channel A* > Threshold 1
1	0 – if Channel A*<= Threshold 2 1 – if Channel A* > Threshold 2
2	0 – if Channel C*<= Threshold 3 1 – if Channel C* > Threshold 3
3	0 – if Channel C*<= Threshold 4 1 – if Channel C* > Threshold 4
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	Pass through mode bit

* value of Channel (A or C) following algorithm which applies pedestal subtraction, weighting factor and drops 2 least significan bits

Table - Format of Data at the Input(address lines) of the Flash RAM Look-up Table

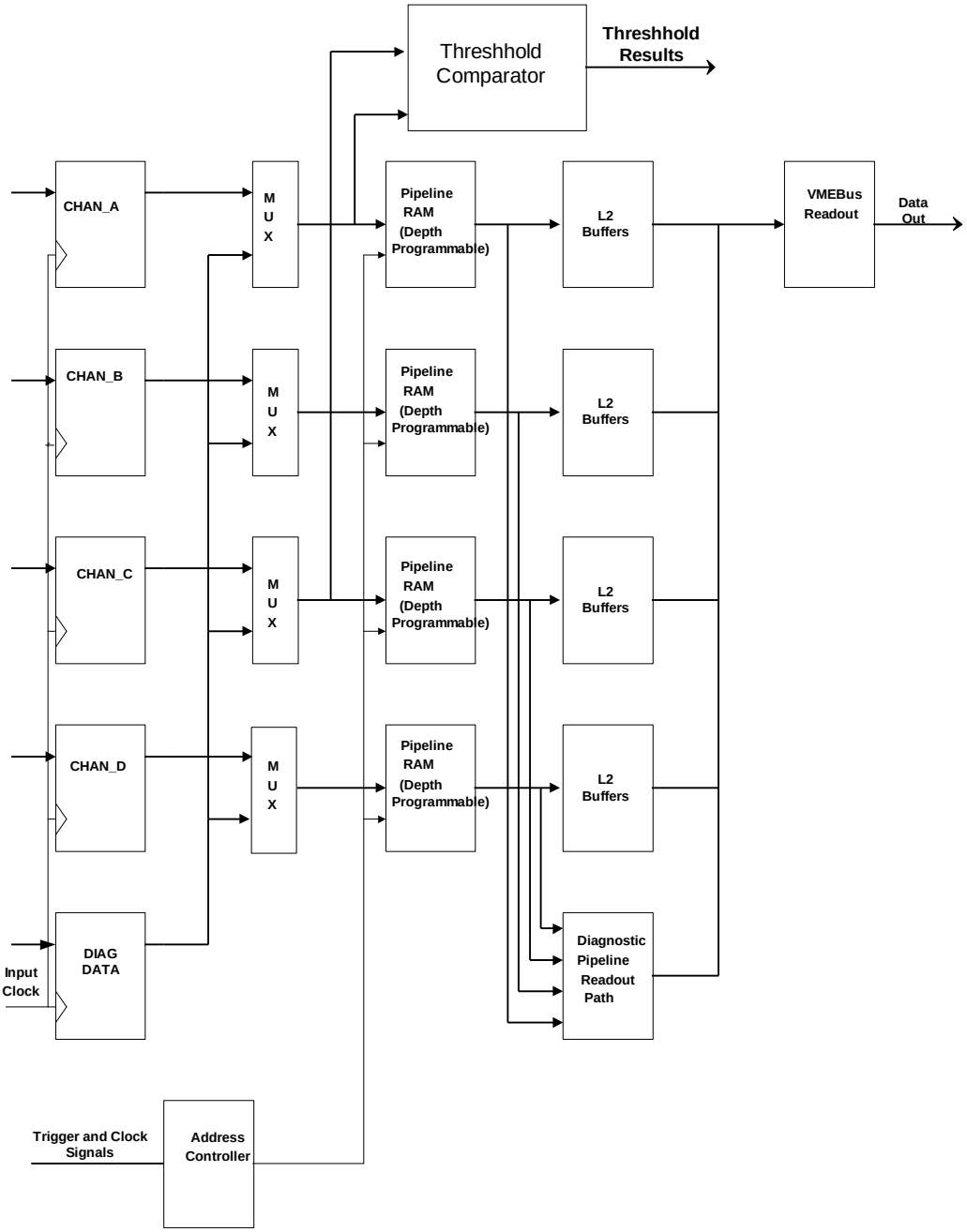
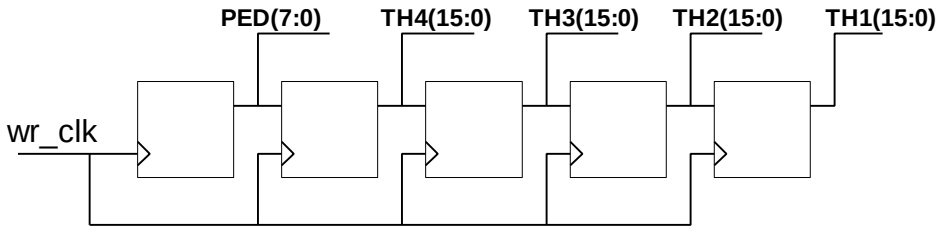


Figure 1 – Block Diagram of TOF Pipeline FPGA

Threshold Register Implementation

The four sixteen bit threshold registers required by the TOF pipeline implementation have been added in the following way.

Because of limited address space, the write path to all these registers is through the pedestal register. A set of 5 registers are chained togeter in a pipelined fahion as shown below:



The last value written to the Pedestal register, will always be the value of the Pedestal register. However, by doing five write cycles, it is possible to load the four Threshhold registers as well as the Pedestal register.

A read of the Pedesal register ALWAYS yields the value last loaded into the Pedestal register. Reads DO NOT effect the Threshold registers, nor is there any way (at present) to read out the threshold registers. It is, of course, always possible to verify their contents by shifting known data through the diagnostic path and checking the result.